

Eduardo Pereira

Universidade de Caxias do Sul
Área de Conhecimento de Exatas e Engenharias
Linguagens Formais – 04/07/25
Segunda Prova

1) (2.5 pontos) Encontre os 5 primeiros conjuntos de itens $LR(0)$ para a gramática a seguir:

$S \rightarrow SaB \mid aB$
 $B \rightarrow bB \mid b$

→ ascendente, recursão à esquerda é permitida

2) (2.5 pontos) Dada a gramática de expressões aritméticas abaixo e a tabela de análise LR, mostre o estado da pilha a cada passo do reconhecimento da sentença $id*(id+id)\$$

- 1) $E \rightarrow E + T$ 4) $T \rightarrow F$
2) $E \rightarrow T$ 5) $F \rightarrow (E)$
3) $T \rightarrow T * F$ 6) $F \rightarrow id$

	id	+	*	()	\$	E	T	F
0	e5			e4			1	2	3
1		e6				AC			
2		r2	e7		r2	r2			
3		r4	r4		r4	r4			
4	e5			e4			8	2	3
5		r6	r6		r6	r6			
6	e5			e4				9	3
7	e5			e4					10
8		e6			e11				
9		r1	e7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

3) (2.5 pontos) Considere a gramática a seguir para declaração de switches simplificados (com apenas um comando em cada entrada) em C (terminais em minúscula, não-terminais em maiúscula):

COMSWITCH \rightarrow switch (id) { LISTA_C }
LISTA_C \rightarrow C LISTA_C | C
C \rightarrow case cte : COM ;
COM \rightarrow break | id = cte

switch *fallthrough*
switch
case
case
break, id

lista C \rightarrow C

Implemente um analisador descendente recursivo preditivo para ela. Faça as alterações que forem necessárias na gramática.

4) (2.5 pontos) Considere a gramática a seguir para declaração de funções em C (terminais em minúscula, não-terminais em maiúscula):

DECLFUNC \rightarrow TIPO id (LISTAPAR) { LISTA_C }
LISTAPAR \rightarrow TIPO id ; LISTAPAR | vazio

TIPO \rightarrow float | int
LISTA_C \rightarrow COM ; LISTA_C | vazio
COM \rightarrow return | id = cte

Implemente um analisador tabular preditivo para ela. Altere a gramática se for necessário.

\rightarrow pois $\text{first}(\text{LISTA-C})$ é $\text{first}(\text{C}())$, que é 'case'

```

int L() {
  if (tk == TKCASE) return LISTA-C();
  return 1;
}
  
```

```

int C() {
  if (tk == TKCASE) {
    gett();
  }
  if (tk == TKCTE) {
    gett();
  }
  if (tk == TKDoisPontos) {
    gett();
  }
  if (COM()) {
    if (tk == TKPontoEVirgula) {
      gett();
      return 1;
    }
  }
  return 0;
}
  
```

```

int COM() {
  if (tk == TKBreak) {
    gett();
    return 1;
  }
  else if (tk == TKID) {
    gett();
    if (tk == TKAtn.b) {
      gett();
    }
    if (tk == TKCTe) {
      gett();
      return 1;
    }
  }
  return 0;
}
  
```

4) DECFUNC = DF
LISTAPAR = LP
TIPO = T
LISTA-C = LC
COM = C

2.5

	first	follow
1) $DF \rightarrow T \text{ id } (LP) \mid LC \mid$	float, int	\$
2) $LP \rightarrow T \text{ id } ; LP \mid \epsilon$	float, int)
3) $LP \rightarrow \epsilon$	ϵ)
4) $T \rightarrow \text{float}$	float	id
5) $T \rightarrow \text{int}$	int	id
6) $LC \rightarrow C ; LC$	return, id	}
7) $LC \rightarrow \epsilon$	ϵ	}
8) $C \rightarrow \text{return}$	return	;
9) $C \rightarrow \text{id} = \text{cte}$	id	;

	id	()	f	}	;	float	int	return	=	cte	\$
DF							1	1				
LP			3				2	2				
T							4	5				
LC	6				7				6			
C	9								8			

int id (float id; int id;) { id = cte; return; }

1. LC para int +

2. LC para float +

3. LC para ; +

4. LC para }